Exact Moment Estimation of Stochastic Differential Dynamics

Shenghua Feng[1], Jie An[1], Naijun Zhan[2], and Fanjiang Xu[1]

[1] Institute of Software, Chinese Academy of Sciences, Beijing, China
[2] School of Computer Science, Peking University, Beijing, China

Abstract. Moment estimation for stochastic differential equations (SDEs) is fundamental to the formal reasoning and verification of stochastic dynamical systems, yet remains challenging and is rarely available in closed form. In this paper, we study time-homogeneous SDEs with polynomial drift and diffusion, and investigate when their moments can be computed exactly. We formalize the notion of moment-solvable SDEs and propose a generic symbolic procedure that, for a given monomial, attempts to construct a finite-dimensional linear ordinary differential equation (ODE) system governing its moment, thereby enabling exact computation. We introduce a syntactic class of pro-solvable SDEs, characterized by a block-triangular structure, and prove that all polynomial moments of any pro-solvable SDE admit such finite ODE representations. This class strictly generalizes linear SDEs and includes many nonlinear models. Experimental results demonstrate the effectiveness of our approach.

Keywords: Stochastic Dynamical Systems, Stochastic Differential Equations (SDEs), Moment Estimation, Pro-solvable SDEs

1 Introduction

Stochastic differential equations (SDEs) are foundational mathematical models for describing the evolution of stochastic systems across a wide range of fields, from physics and biology [21] to finance [3] and engineering [12]. Moments, defined as expectations of monomial functions of the system variables, provide critical insights into the behavior and stability of the underlying stochastic systems. Consequently, moment estimation is a central problem in various applications, including formal verification of stochastic dynamical systems [7,15,2], sensitivity analysis [1,10], and the derivation of bounds for safety-critical models [23,18].

Despite their importance, exact moment estimation for SDEs remains a formidable challenge. Since SDEs, even linear ones, are often difficult to solve in closed form, their moments are correspondingly hard to compute directly. To the best of our knowledge, existing approaches based on the synthesis of supermartingales [27] can provide bounds on moments [22], but these methods are often conservative and rely heavily on the choice of template functions for the supermartingale, which may require significant manual effort. As illustrated in our case study (cf. Sect. 5.1), such approaches may fail to yield tight estimates for the moments of interest. Currently, there is no general framework for the exact computation of moments in general SDEs.

In this work, we focus on time-homogeneous SDEs with polynomial drift and diffusion terms, and seek to systematically characterize when the moments can be computed exactly. Motivated by advances in the verification of probabilistic programs [2], we introduce the notion of *moment-solvable* SDEs, for which every moment admits an explicit, closed-form solution. Central to our approach is a symbolic procedure that, given any monomial, attempts to construct a finite-dimensional linear ODE system governing the moment dynamics. This procedure iteratively expands the set of coupled moments by applying the infinitesimal generator of the SDE, and halts if the expansion closes after finitely many steps.

Our main theoretical contribution is the identification of a syntactic class of SDEs, termed *pro-solvable SDEs*, which are characterized by a block-triangular structure in their coefficients. We prove that for all pro-solvable SDEs, the symbolic closure procedure always terminates, ensuring that every moment can be computed by solving a finite-dimensional linear ODE system. Notably, this class strictly generalizes the linear SDEs, encompassing a broad array of nonlinear models encountered in practice.

We demonstrate the practical effectiveness of our approach through experiments on a diverse suite of SDE benchmarks. Our method efficiently computes exact moments for many linear and nonlinear systems of interest, including higher-order cases, illustrating its scalability and applicability to the formal analysis of stochastic systems.

In summary, our main contributions are as follows:

– After the problem formulation (Sect. 2), we introduce the concept of moment-solvable SDEs and establish a general procedure for constructing finite-dimensional ODE systems governing the moment dynamics. (Sect. 3)
– We introduce and characterize the class of pro-solvable SDEs, proving that they are moment-solvable and that all their moments can be computed exactly. Furthermore, we provide a complexity analysis of our method. (Sect. 4)
– We implement our method and conduct experiments to demonstrate the broad applicability and effectiveness of our approach. (Sect. 5)

Related work. The closest related works are those on *prob-solvable loops* and related classes of probabilistic programs in discrete time [2,18], where loop moments satisfy solvable linear recurrences and can be computed exactly. Our setting differs in that we consider continuous-time SDEs and build on the infinitesimal generator. A second line of work provides structural exact-moment results for stochastic reaction networks [17] and jump Markov processes [26,4], where specific network topologies (i.e., feedforward structures) yield closed finite moment equations. These results, however, are tailored to particular classes of jump processes [9,25] and do not offer a uniform procedure for general polynomial SDEs. Finally, existing works employ martingales [22,5,11] and semidefinite relaxations [8,16] to approximate moments and provide bounds used for the verification and analysis of SDEs and stochastic hybrid systems.

2 Problem Formulation

Let $\mathbb{N}$, $\mathbb{Z}$, and $\mathbb{R}$ denote the sets of natural numbers, integers, and real numbers, respectively. Vectors are denoted in bold; for $\boldsymbol{x} \in \mathbb{R}^n$, x_i refers to its i-th component. For $\boldsymbol{x} = (x_1, \ldots, x_n)^\top$, let $\alpha \triangleq (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$ be a multi-index, with $|\alpha| \triangleq \sum_{i=1}^n \alpha_i$. We use the notation $\boldsymbol{x}^\alpha \triangleq x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ for presenting the corresponding monomial.

Probability and Moments. Let $(\Omega, \mathcal{F}, P)$ be a probability space, where Ω is the sample space, $\mathcal{F} \subseteq 2^\Omega$ is a σ-algebra, and $P \colon \mathcal{F} \to [0,1]$ is a probability measure. A *random variable* X defined on $(\Omega, \mathcal{F}, P)$ is an $\mathcal{F}$-measurable function $X \colon \Omega \to \mathbb{R}^n$; its expectation (w.r.t. P) is denoted by $\mathbb{E}[X]$. Given a multi-index $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$, the α-moment of X is $\mathbb{E}[X^\alpha] \triangleq \mathbb{E}[(X^{(1)})^{\alpha_1} \cdots (X^{(n)})^{\alpha_n}]$, where $X^{(i)}$ denotes the i-th component of X. A (continuous-time) *stochastic process* is a collection of random variables $\{X_t\}_{t \in T}$, where unless otherwise noted, the index set T is the half-line $[0, \infty)$.

Stochastic Differential Equations (SDEs). We consider a class of stochastic dynamical systems governed by time-homogeneous stochastic differential equations (SDEs) of the form

$$dX_t = b(X_t)\, dt + \sigma(X_t)\, dW_t, \quad t \geq 0, \tag{1}$$

where $\{X_t\}$ is an n-dimensional continuous-time stochastic process, $\{W_t\}$ is an m-dimensional Wiener process (standard Brownian motion), $b \colon \mathbb{R}^n \to \mathbb{R}^n$ is a vector-valued polynomial drift coefficient modeling the deterministic part of the dynamics, and $\sigma \colon \mathbb{R}^n \to \mathbb{R}^{n \times m}$ is a matrix-valued polynomial diffusion coefficient encoding the system's coupling to Gaussian white noise dW_t.

Under standard regularity and growth conditions [19, Chap. 5.2], given an initial state (random variable) X_0, the SDE (1) admits a unique solution $X_t(\omega) = X(t, \omega) \colon [0, \infty) \times \Omega \to \mathbb{R}^n$ that satisfies the stochastic integral equation

$$X_t = X_0 + \int_0^t b(X_s)\, ds + \int_0^t \sigma(X_s)\, dW_s.$$

The solution process $\{X_t\}$ of (1) is also called an *(Itô) diffusion process*, and may be denoted X_t^{0,X_0} (or simply $X_t^{X_0}$) to indicate the initial condition X_0 at time $t = 0$. In the special case where $\sigma \equiv 0$, the SDE reduces to an ordinary differential equation (ODE), recovering the classical deterministic setting.

The *exact moment estimation* (EME) problem of SDEs studied in this paper reads as follows:

EME Problem. Let $\{X_t\}_{t \geq 0}$ denote the solution of SDE (1). Our objective is to compute the α-moment of the random variable X_t, that is,

$$m_\alpha(t) \;\triangleq\; \mathbb{E}[X_t^\alpha] = \mathbb{E}\big[\big(X_t^{(1)}\big)^{\alpha_1} \cdots \big(X_t^{(n)}\big)^{\alpha_n}\big]$$

for any given multi-index $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$ and any $t \geq 0$.

3 Reduction of EME Problem to ODE Solving

In this section, we formalize the concept of *moment-solvable* SDEs, which precisely delineates those systems for which the exact moment estimation problem admits an explicit solution. We then present a generic symbolic procedure that, given a target monomial, systematically attempts to construct a finite-dimensional linear ODE system governing the evolution of its moment, thus enabling exact computation of the desired expectation.

Definition 1 (Moment-solvable SDE). *Given a multi-index $\alpha \in \mathbb{N}^n$, we say that the SDE (1) is* moment-solvable for α *if there exists an explicit function $h_\alpha \colon [0, \infty) \to \mathbb{R}$ such that*

$$m_\alpha(t) = \mathbb{E}[X_t^\alpha] = h_\alpha(t) \quad \textit{for all } t \geq 0.$$

If this property holds for all multi-indices α, then the SDE is moment-solvable*.*

Clearly, solving the EME problem for a multi-index α is equivalent to establishing that the SDE is moment-solvable for α. In practice, obtaining explicit formulas for moments is highly nontrivial. First, closed-form solutions for nonlinear SDEs are generally unavailable. Second, even when a stochastic representation of the solution is known, the evaluation of $\mathbb{E}[X_t^\alpha]$ typically involves high-dimensional integrals that do not admit simple analytical expressions.

To circumvent these challenges, rather than attempting to compute $\mathbb{E}[X_t^\alpha]$ directly, we instead consider the time evolution of moments. By deriving differential equations for moments using the infinitesimal generator, we seek to construct a closed, finite-dimensional linear ODE system for the evolution of a suitable collection of moments. This approach is grounded in Dynkin's formula, which serves as the stochastic analogue of the Newton–Leibniz rule and connects the dynamics of the process to the evolution of expected values.

Theorem 1 (Dynkin's formula [19]). *Let $\{X_t\}_{t\geq 0}$ be the solution of (1). If $f \in C^2(\mathbb{R}^n)$ has compact support, then for all $t \geq 0$,*

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[f(X_t)] = \mathbb{E}[(\mathcal{A}f)(X_t)],$$

where $\mathcal{A}$ is the infinitesimal generator of (1) given by

$$\mathcal{A}f(\boldsymbol{x}) = \sum_{i=1}^{n} b_i(\boldsymbol{x}) \frac{\partial f}{\partial x_i}(\boldsymbol{x}) + \frac{1}{2} \sum_{i,j=1}^{n} (\sigma\sigma^\top)_{ij}(\boldsymbol{x}) \, \frac{\partial^2 f}{\partial x_i \partial x_j}(\boldsymbol{x}).$$

Since both the drift vector $b(\boldsymbol{x})$ and the diffusion matrix $\sigma(\boldsymbol{x})$ are polynomial, applying Dynkin's formula to the test function $f(\boldsymbol{x}) = \boldsymbol{x}^\alpha$ yields

$$\begin{aligned} \frac{d}{dt} m_\alpha(t) \; &= \; \mathbb{E}[\mathcal{A}(\mathbf{x}^\alpha)|\mathbf{x} = X_t] \qquad\qquad [\text{Suppose } \mathcal{A}(\boldsymbol{x}^\alpha) = \sum_\gamma a_{\alpha\gamma}\boldsymbol{x}^\gamma + c_\alpha \;] \\ &= \; \mathbb{E}\left[\sum_\gamma a_{\alpha\gamma} X_t^\gamma\right] + c_\alpha = \sum_\gamma a_{\alpha\gamma} \mathbb{E}[X_t^\gamma] + c_\alpha = \sum_\gamma a_{\alpha\gamma} m_\gamma(t) + c_\alpha \qquad (2) \end{aligned}$$

Algorithm 1 Construction of a finite moment system for a given multi-index α

Require: Drift b, diffusion σ, generator $\mathcal{A}$ of (1), initial multi-index α
Ensure: multi-index set S which entails a closed linear ODE of moments if terminates.

```
1: S ← {α}                                        ▷ set of multi-indices (monomials)
2: P ← {α}                                        ▷ set of unprocessed multi-indices
3: while P ≠ ∅ do
4:     Select and remove some β ∈ P
5:     Compute the generator action on the monomial x^β: A x^β = Σ_γ a_{βγ} x^γ + c_β
6:     for all γ such that a_{βγ} ≠ 0 do
7:         if γ ∉ S then
8:             S ← S ∪ {γ}
9:             P ← P ∪ {γ}
10: Let S = {α, β^1, ..., β^k} be the final set of indices.
11: Form the moment vector m(t) ← (E[X_t^α], E[X_t^{β_1}], ..., E[X_t^{β_k}])^⊤.
12: Use Thm. 1 to obtain the closed linear ODE system d/dt m(t) = A m(t) + c.
```

This procedure reveals that the dynamics of any single moment are coupled to the dynamics of other moments. We can systematically uncover the full set of coupled moments by starting with our initial moment m_α and recursively applying the generator $\mathcal{A}$ to any new monomials that appear on the right-hand side of Eq. (2).

We formalize this process by constructing a set of monomials that is closed under the action of $\mathcal{A}$, as described in Alg. 1 (the *moment closure algorithm*). The key steps are summarized as follows:

1. **Initialization.** Initialize the set of monomials $S := \{\boldsymbol{x}^\alpha\}$.
2. **Closure construction.** While there exists a monomial $\boldsymbol{x}^\beta \in S$ that has not yet been processed, compute $\mathcal{A}(\boldsymbol{x}^\beta)$ and expand it as a linear combination of monomials (line 4-5 in Alg. 1):
$$\mathcal{A}\boldsymbol{x}^\beta = \sum_\gamma a_{\beta\gamma}\, \boldsymbol{x}^\gamma + c_\beta.$$
For each new monomial $\boldsymbol{x}^\gamma$ appearing on the right-hand side that is not already in S, add $\boldsymbol{x}^\gamma$ to S. Mark $\boldsymbol{x}^\beta$ as processed. (line 6-9 in Alg. 1)
3. **Exact moment calculation.** If this procedure terminates after finitely many steps, we obtain a finite multi-index set $S = \{\alpha, \beta_1, \ldots, \beta_k\}$. In this case, defining the moment vector $m(t) \triangleq \big(\mathbb{E}[X_t^\alpha], \mathbb{E}[X_t^{\beta_1}], \ldots, \mathbb{E}[X_t^{\beta_k}]\big)^\top$, Dynkin's formula yields a closed linear ODE system
$$\frac{\mathrm{d}}{\mathrm{d}t} m(t) = A\, m(t) \,+\, c\,, \tag{3}$$
where the matrix A and the vector c collects the coefficients from the generator expansions.

If Alg. 1 terminates, the resulting finite-dimensional ODE system yields explicit expressions for all moments in the set S, and in particular for the target moment $\mathbb{E}[X_t^\alpha]$. Consequently, whenever the closure procedure terminates, the SDE is moment-solvable for α. We formalize it as the following theorem.

Theorem 2 (Explicit moment computation). *If Alg. 1 terminates, then SDE (1) is moment-solvable for α. In particular, the moment vector $m(t)$ can be explicitly expressed as*

$$\left(\mathbb{E}[X_t^\alpha], \mathbb{E}[X_t^{\beta_1}], \ldots, \mathbb{E}[X_t^{\beta_k}]\right)^\top \;=\; m(t) \;=\; e^{At} m(0) + \int_0^t e^{A(t-s)}\, c \;\mathrm{d}s. \tag{4}$$

where $m(0) = \left(\mathbb{E}[X_0^\alpha],\, \mathbb{E}[X_0^{\beta_1}],\, \ldots,\, \mathbb{E}[X_0^{\beta_k}]\right)^\top$ is determined by the initial distribution, and the target moment $\mathbb{E}[X_t^\alpha]$ is given by the first component of $m(t)$.

Proof. Since Alg. 1 terminates, we obtain a linear ODE system Eq. (3) for $m(t)$, whose solution is given explicitly by Eq. (4). This completes the proof. $\square$

Remark 1. The matrix exponential and integral appearing on the right-hand side of (4) can typically be evaluated in closed form using standard symbolic computation tools (e.g., MATHEMATICA), making the explicit computation of $m(t)$ readily achievable in practice.

Depending on the specific moment m_α and the structure of the drift and diffusion coefficients, Alg. 1 may either terminate or diverge. The following example illustrates a nonlinear dynamics for which the closure procedure terminates, resulting in a finite-dimensional moment system.

Example 1 (Chemical process in an Ornstein–Uhlenbeck environment [13]). Consider a chemical system, where X_t models a fluctuating environment and Y_t denotes the concentration of a chemical species influenced by the environment:

$$\begin{cases} \mathrm{d}X_t = -X_t \;\mathrm{d}t + \mathrm{d}W_t^{(1)}, \\ \mathrm{d}Y_t = \left(-2Y_t + X_t + X_t^2\right) \mathrm{d}t + X_t \;\mathrm{d}W_t^{(2)}, \end{cases} \tag{5}$$

Suppose the initial state is $(X_0, Y_0) = (0, 0)$. We seek to compute the second moment of Y_t, i.e., $m_{(0,2)} = \mathbb{E}[Y_t^2]$. Applying Alg. 1, we obtain a closed 8-dimensional linear ODE system for the collection of moments

$$m_{(0,2)}, \quad m_{(2,1)}, \quad m_{(2,0)}, \quad m_{(1,1)}, \quad m_{(4,0)}, \quad m_{(3,0)}, \quad m_{(0,1)}, \quad m_{(1,0)}$$

where $m_{(i,j)} \triangleq \mathbb{E}[X_t^i Y_t^j]$. Solving this system (see Appendix A for the explicit ODE system), we obtain an explicit formula for the second moment:

$$\mathbb{E}\left[Y_t^2\right] = \frac{1}{3} + \frac{2}{3}e^{-3t} + \left(-\frac{t}{4} - \frac{11}{8}\right) e^{-2t} + \left(\frac{3}{4}t^2 + t + \frac{3}{8}\right) e^{-4t}. \qquad \triangleleft$$

The following example illustrates a case in which Alg. 1 does *not* terminate.

Example 2 (Double-well potential [6]). Consider the bistable Langevin system describing the dynamics of a particle in a double-well potential,

$$\mathrm{d}X_t = \left(aX_t - X_t^3\right) dt + \sigma\, dW_t,$$

where $a, \sigma \in \mathbb{R}$ are constants. Applying the generator $\mathcal{A}$ to the monomial x^n for $n \geq 2$, we obtain $\mathcal{A}x^n = -nx^{n+2} + anx^n + \frac{\sigma^2}{2}n(n-1)x^{n-2}$. Therefore, starting from x^n, each application of the generator introduces a new, higher-degree monomial x^{n+2}, and recursively, all monomials of the form x^{n+2k} for $k \geq 0$ are generated. As a result, the set S continues to expand indefinitely, and Alg. 1 does not terminate. $\triangleleft$

Combining Thm. 2 with Def. 1, we obtain the following characterization:

Theorem 3. *Given an SDE* (1)*, if Alg. 1 terminates for every monomial* $\boldsymbol{x}^\alpha$*, then the SDE* (1) *is moment-solvable; that is, explicit closed-form expressions can be computed for all moments.*

4 A Class of SDEs with Moment-Solvable Property

In this section, we identify a class of SDEs, termed *pro-solvable SDEs*, for which Alg. 1 terminates for any monomial $\boldsymbol{x}^\alpha$, thereby ensuring the moment-solvable property by Thm. 3. The class of pro-solvable SDEs contains both linear SDEs and certain nonlinear SDEs whose variables exhibit a triangular dependence structure. We present the definition of pro-solvable SDEs in Sect. 4.1 and prove Alg. 1 terminates for any pro-solvable SDE and any multi-index α in Sect. 4.2.

4.1 Pro-Solvable SDEs

We begin by formalizing the concept of *ordered partition*, which serves as a foundation for introducing the notion of pro-solvable SDEs.

Definition 2 (Ordered partition). *We say that non-empty blocks* $G_1, G_2, \ldots, G_r$ *form an* ordered partition *of* $\{1, 2, \ldots, n\}$ *if*

$$\{1, 2, \ldots, n\} = G_1 \sqcup \cdots \sqcup G_r, \qquad G_i \cap G_j = \emptyset \quad \text{for any } i \neq j,$$

and the blocks $G_1, G_2, \ldots, G_r$ *are equipped with the natural order* $G_1 \prec \cdots \prec G_r$*. For each* $p \in \{1, \ldots, r\}$*, let* $\boldsymbol{x}^{(p)} \triangleq (x_i)_{i \in G_p}$ *denote the collection of variables whose indices belong to block* G_p*, and* $\boldsymbol{x}^{(<p)} \triangleq (x_i)_{i \in G_1 \cup \cdots \cup G_{p-1}}$ *be the collection of all variables whose indices belong to the preceding blocks of* G_p*.*

Definition 3 (Pro-solvable SDEs). *An SDE* (1) *is* pro-solvable *if there exists an ordered partition* $G_1, G_2, \ldots, G_r$ *of* $\{1, 2, \ldots, n\}$*, such that its drift* $b = (b_1, \ldots, b_n)$ *and diffusion matrix* $\sigma = (\sigma_{ik})_{1 \leq i \leq n, 1 \leq k \leq m}$ *satisfy the* block-triangular affine *structure: for every block* G_p*, and for all* $i \in G_p$*,* $k = 1, \ldots, m$*,*

$$b_i(\boldsymbol{x}) = \sum_{j \in G_p} \Lambda_{ij} x_j + B_i\big(x^{(<p)}\big), \qquad \Lambda_{ij} \in \mathbb{R}, \ \ B_i \in \mathbb{R}[\boldsymbol{x}^{(<p)}], \tag{6}$$

$$\sigma_{ik}(\boldsymbol{x}) = \sum_{j \in G_p} A_{ikj} x_j + P_{ik}\big(x^{(<p)}\big), \qquad A_{ikj} \in \mathbb{R}, \ \ P_{ik} \in \mathbb{R}[\boldsymbol{x}^{(<p)}]. \tag{7}$$

In particular, each b_i *and* σ_{ik} *is affine-linear in the variables of its own block* $\boldsymbol{x}^{(p)}$*, and any nonlinearity depends only on variables from earlier blocks* $\boldsymbol{x}^{(<p)}$*.*

The block-triangular affine structure in pro-solvable SDEs imposes a natural hierarchy among the variables: within each block G_p, the drift and diffusion coefficients are affine-linear functions of the variables in that block, while any nonlinear dependence is restricted to variables in preceding blocks $\boldsymbol{x}^{(<p)}$. It ensures that the moment dynamics associated with variables in higher-indexed blocks only depend on moments of lower-indexed blocks, and never vice versa.

Example 3. Reconsider the SDE in Exmp. 1, given in Eq. (5). Suppose its drift and diffusion coefficients are

$$b(x_1, x_2) = \begin{pmatrix} -x_1 \\ -2x_2 + x_1 + x_1^2 \end{pmatrix}, \qquad \sigma(x_1, x_2) = \begin{pmatrix} 1 & 0 \\ 0 & x_1 \end{pmatrix}.$$

It is straightforward to verify that SDE (5) is pro-solvable under the ordered partition $G_1 = \{1\}$, $G_2 = \{2\}$ with $r = 2$. $\triangleleft$

The class of pro-solvable SDEs subsumes several important subclasses. In particular, when the ordered partition is taken as (i) $r = 1$ with $G_1 = \{1, 2, \dots, n\}$, the pro-solvable SDEs specializes to the well-known class of linear SDEs, in which both the drift and diffusion coefficients are affine functions of all variables. On the other hand, when the partition is (ii) $r = n$ with $G_i = \{i\}$ for $1 \le i \le n$, the pro-solvable condition reduces to the strictly triangular case, where each variable may depend nonlinearly only on those variables with strictly smaller indices.

4.2 Pro-solvable SDEs are moment-solvable

In this subsection, we establish that pro-solvable SDEs guarantee termination of the iterative moment closure procedure described in Alg. 1 (Lines 3–9). That is, for any monomial $\boldsymbol{x}^\alpha$, Alg. 1 generates only finitely many new moments, thereby ensuring the moment-solvable property.

Clearly, the termination of Alg. 1 hinges on whether infinitely many new monomials are added to the set $\mathcal{P}$. The intuitive strategy is to construct a ranking function over monomials such that, whenever a new monomial $\boldsymbol{x}^\gamma$ arises from the generator action $\mathcal{A}\boldsymbol{x}^\beta$, its rank is no greater than that of $\boldsymbol{x}^\beta$.

To formalize this idea, we examine in detail how the operator $\mathcal{A}$ acts on pro-solvable SDEs. For such systems, the generator $\mathcal{A}$ expands as

$$\begin{aligned}
\mathcal{A} f &= \sum_{i=1}^{n} b_i(\boldsymbol{x}) \partial_{x_i} f + \frac{1}{2} \sum_{i,j=1}^{n} \sum_{k=1}^{m} \sigma_{ik}(\boldsymbol{x}) \sigma_{jk}(\boldsymbol{x}) \, \partial_{x_i} \partial_{x_j} f \\
&= \sum_{i=1}^{n} \left(\sum_{l \in G_{I(i)}} \Lambda_{il} x_l + B_i\big(\boldsymbol{x}^{(<I(i))}\big) \right) \partial_{x_i} f \qquad (8) \\
&+ \frac{1}{2} \sum_{i,j=1}^{n} \sum_{k=1}^{m} \left(\sum_{l \in G_{I(i)}} A_{ikl} x_l + P_{ik}\big(\boldsymbol{x}^{(<I(i))}\big) \right) \left(\sum_{l \in G_{I(j)}} A_{jkl} x_l + P_{jk}\big(\boldsymbol{x}^{(<I(j))}\big) \right) \partial_{x_i} \partial_{x_j} f
\end{aligned}$$

where $I(i)$ denotes the unique block index for which i belongs to $G_{I(i)}$, and ∂_{x_i} is the partial differential operator that maps a function f to ${}^{\partial f}/_{\partial x_i}$. Thus, the operator $\mathcal{A}$ is a linear combination of primitive terms of the form

$$\boldsymbol{x}^{\gamma}\,\partial_{x_i}, \qquad \boldsymbol{x}^{\gamma}\,\partial^2_{x_i}, \qquad \boldsymbol{x}^{\gamma}\,\partial_{x_i}\partial_{x_j}, \tag{9}$$

where the monomial $\boldsymbol{x}^{\gamma}$ arises from the polynomial coefficients of b_i and σ_{ij}. Depending on how $\boldsymbol{x}^{\gamma}$ is produced, we distinguish two types of primitive terms.

Definition 4 (Classification of primitive terms in $\mathcal{A}$). *Suppose SDE (1) is pro-solvable with ordered partition $G_1, G_2, \ldots, G_r$, then the primitive terms in operator $\mathcal{A}$ are classified into*

(i) Linear-produced terms*: if monomial $\boldsymbol{x}^{\gamma}$ in primitive term comes entirely from the affine linear parts $\sum_{l\in G_p} \Lambda_{il}x_l$, $\sum_{l\in G_p} A_{ikl}x_l$ or $\sum_{l\in G_q} A_{jkl}x_l$ with no factor from any B_i, P_{ik} or P_{jk}. A monomial in linear-produced terms is not necessarily linear, as we allow the product of two affine linear parts.*

(ii) Polynomial-produced terms*: if monomial $\boldsymbol{x}^{\gamma}$ in primitive term contains at least one factor from some $B_i(\boldsymbol{x}^{(<I(i))})$, $P_{ik}(\boldsymbol{x}^{(<I(i))})$, or $P_{jk}(\boldsymbol{x}^{(<I(j))})$.*

For a polynomial-produced primitive term, we define its *source block index* as follows. If the monomial $\boldsymbol{x}^{\gamma}$ contains a factor from B_i or P_{ik} for some $i \in G_p$, then p is regarded as a candidate source block index. In cases where multiple candidate source block indices exist (e.g., cross terms involving both P_{ik} and P_{jk}), we select the source block index as the larger one (e.g. $\max\{p, q\}$ if $i \in G_p$ and $j \in G_q$ in the cross term case). Consequently, every polynomial-produced primitive term is associated with a unique source block index $p \in \{1, \ldots, r\}$.

Example 4. Continuing Exmp. 3, the generator $\mathcal{A}$ corresponding to SDE (5) simplifies to

$$\mathcal{A} \;=\; -\,x_1\,\partial_{x_1} + \left(-2x_2 + x_1 + x_1^2\right)\partial_{x_2} + \frac{1}{2}\,\partial^2_{x_1} + \frac{1}{2}x_1^2\,\partial^2_{x_2}. \tag{10}$$

Following the classification in Def. 4, the linear-produced terms are $\{x_1\,\partial_{x_1}, x_2\,\partial_{x_2}\}$, while the polynomial-produced terms are $\{x_1\,\partial_{x_2}, x_1^2\,\partial_{x_2}, \partial^2_{x_1}, x_1^2\partial^2_{x_2}\}$. Moreover, the corresponding source block indices for the polynomial-produced terms $x_1\,\partial_{x_2}$, $x_1^2\,\partial_{x_2}$, $\partial^2_{x_1}$, and $x_1^2\,\partial^2_{x_2}$ are 2, 2, 1, and 2 respectively. ◁

We proceed to analyze how the exponents change when applying $\mathcal{A}$ to a monomial $\boldsymbol{x}^{\beta}$. To facilitate this analysis, we introduce the notion of the *block exponent sum* and *block exponent difference*.

Definition 5. *Given the notations above, let β be a multi-index with $\boldsymbol{x}^{\beta} = \prod_{i=1}^{n} x_i^{\beta_i}$. For each block G_p, the block exponent sum is defined as*

$$s_p(\beta) \;\triangleq\; \sum_{i\in G_p} \beta_i, \qquad \textit{for } p = 1, \ldots, r.$$

Moreover, for any two multi-indices β and β', the block exponent difference is defined by $\Delta s_p(\beta, \beta') \triangleq s_p(\beta') - s_p(\beta)$ for $1 \le p \le r$.

Now, given a monomial $\boldsymbol{x}^{\beta}$, let $\boldsymbol{x}^{\beta'}$ be any monomial that appears in the expansion of $\mathcal{A}\boldsymbol{x}^{\beta}$. According to Eq. (8), there must exist a unique primitive term T of the form $\boldsymbol{x}^{\gamma}\,\partial_{x_i}$, $\boldsymbol{x}^{\gamma}\,\partial_{x_i}^2$, or $\boldsymbol{x}^{\gamma}\,\partial_{x_i}\partial_{x_j}$ such that $T\boldsymbol{x}^{\beta} = c\,\boldsymbol{x}^{\beta'}$ for some constant c. The following result shows that the change in exponents is bounded, and this bound is independent of the specific monomials $\boldsymbol{x}^{\beta}$ and $\boldsymbol{x}^{\beta'}$.

Lemma 1 (Bound on exponent change). *Given a monomial $\boldsymbol{x}^{\beta}$, let $\boldsymbol{x}^{\beta'}$ be any monomial that appears in the expansion of $\mathcal{A}\boldsymbol{x}^{\beta}$. Suppose $\boldsymbol{x}^{\beta'}$ is produced by a primitive term T (i.e., T is of the form given in Eq. (9), and $T\boldsymbol{x}^{\beta} = c\,\boldsymbol{x}^{\beta'}$ for some constant c). Then the following properties hold:*

- *If T is a linear-produced term, then $\Delta s_q(\beta,\beta') \le 0$ for all $q = 1,\dots,r$.*
- *If T is a polynomial-produced term with source block index p, then*
  (1) $\Delta s_p(\beta,\beta') \le -1$;
  (2) $\Delta s_q(\beta,\beta') = 0$ for all $q > p$;
  (3) For each $q < p$, there exists a constant $C_{p,q} \in \mathbb{N}$ (independent of β and β', depending only on q and on the degrees of B_i and P_{ik} for all i in $G_1 \cup \cdots \cup G_p$) such that $0 \le \Delta s_q(\beta,\beta') \le C_{p,q}$.

Proof. The proof proceeds by a case-by-case analysis of the primitive term T.

Case 1: T is a linear-produced term. Suppose T is a linear-produced term, then T can only take the form in either $x_l\,\partial_{x_i}$ for some $i, l \in G_p$ or $x_l x_{l'}\,\partial_{x_i}\partial_{x_j}$ for some $i, l \in G_p$ and $j, l' \in G_{p'}$. A direct calculation shows that $\Delta s_q \le 0$ for all $q = 1,\dots,r$. Indeed, each derivative ∂_{x_i} reduces the exponent of some x_i in its block by 1, and the linear coefficient can reintroduce at most as many variables in that block as there are derivatives; hence the total exponent in each block never increases under linear-produced terms.

Case 2: T is a polynomial-produced term. Suppose T is a polynomial-produced term with source block index p. By Eq. (8) and definition of the source block index, T always contains at least one derivative with respect to some x_i in block G_p, and T can only take the form in the following three cases:

* $\boldsymbol{x}^{\gamma}\partial_{x_i}$ with $\boldsymbol{x}^{\gamma}$ a monomial in $B_i(\boldsymbol{x}^{(<p)})$;
* $\boldsymbol{x}^{\gamma}\partial_{x_i}^2$ with $\boldsymbol{x}^{\gamma}$ a monomial in $2x_l P_{ik}(\boldsymbol{x}^{(<p)})$ or $\big(P_{ik}(\boldsymbol{x}^{(<p)})\big)^2$ for some k and some $l \in G_p$;
* $\boldsymbol{x}^{\gamma}\partial_{x_i}\partial_{x_j}$ with $\boldsymbol{x}^{\gamma}$ a monomial in $P_{ik}(\boldsymbol{x}^{(<p)})P_{jk}(\boldsymbol{x}^{(<u)})$ for some k, and for some $j \ne i$ with $j \in G_u$ and $u \le p$.

In either case, we can directly check that:

(1) $\Delta s_p \le -1$, i.e. the total exponent in the source block G_p strictly decreases;
(2) $\Delta s_q = 0$ for all $q > p$, i.e. no later block is affected;
(3) For each $q < p$, there exists a constant $C_{p,q} \in \mathbb{N}$ (*depending only on* q and on the degrees of B_i and P_{ik} for all i in $G_1 \cup \cdots \cup G_p$) such that $0 \le \Delta s_q \le C_{p,q}$.

Intuitively, the above result implies each application of a polynomial-produced term from block p differentiates at least once in some variable of G_p, thus reducing s_p by one or two, while the coefficient can at most reintroduce at most one variable from G_p and some bounded amount of variables from earlier blocks $G_1,\dots,G_{p-1}$. Combining both cases, we obtain the desired result. □

At the beginning of this subsection, we mention that our strategy is to construct a ranking function over monomials to prove the termination of Alg. 1. Now, we introduce the following weighted block degree serving as a ranking function, such that its value over a newly added monomial does not increase.

Definition 6. *Given the notations above, the* weighted block degree *for a monomial is defined as*

$$\deg_W(\boldsymbol{x}^\beta) \;\triangleq\; \sum_{p=1}^{r} W_p\, s_p(\beta),$$

where the block weights $W_1, \ldots, W_r$ are chosen as follows: set $W_1 \triangleq 1$, and for each $p = 2, \ldots, r$, choose W_p inductively by

$$W_p \;>\; \sum_{q<p} C_{p,q}\, W_q.$$

The following lemma shows that the weighted block degree is indeed non-increasing. Intuitively, by Lem. 1, no new higher-degree monomials are created within the same block. Any additional complexity can only arise from dependencies on variables in earlier blocks, whose closure is handled inductively. This behavior is precisely captured by the choice of block weights in Def. 6.

Lemma 2 (Non-increase of weighted block degree). *Given a monomial $\boldsymbol{x}^\beta$, let $\boldsymbol{x}^{\beta'}$ be any monomial that appears in the expansion of $\mathcal{A}\boldsymbol{x}^\beta$, then*

$$\deg_W(\boldsymbol{x}^{\beta'}) - \deg_W(\boldsymbol{x}^\beta) \;\leq\; 0\,.$$

Proof. Given the notations above, suppose T is a polynomial-produced term with source block p, then by Lem. 1

$$\deg_W(\boldsymbol{x}^{\beta'}) - \deg_W(\boldsymbol{x}^\beta) = \sum_{q=1}^{r} W_q\, \Delta s_q \;\leq\; W_p * (-1) + \sum_{q<p} W_q C_{p,q} \;<\; 0.$$

Therefore, any genuinely new monomial produced by a polynomial-produced term strictly *decreases* the weighted block degree. On the other hand, If T is a linear-produced term, we have, by Lem. 1

$$\deg_W(\boldsymbol{x}^{\beta'}) - \deg_W(\boldsymbol{x}^\beta) = \sum_{q=1}^{r} W_q\, \Delta s_q \;\leq\; 0,$$

Hence, the weighted degree is *nonincreasing* under linear-produced terms. This completes the proof. □

Consider the directed graph whose vertices are monomials and with an edge $\boldsymbol{x}^\beta \to \boldsymbol{x}^{\beta'}$ whenever $\boldsymbol{x}^{\beta'}$ appears in $\mathcal{A}\boldsymbol{x}^\beta$. Starting from $\boldsymbol{x}^\alpha$, any monomial that appears in some $\mathcal{A}^m(x^\alpha)$ is reachable via a directed path. Along any path of distinct vertices $\boldsymbol{x}^{\beta^{(0)}} = \boldsymbol{x}^\alpha \to \boldsymbol{x}^{\beta^{(1)}} \to \boldsymbol{x}^{\beta^{(2)}} \to \cdots$, the sequence $\deg_W(\boldsymbol{x}^{\beta^{(k)}})$ is non-increasing by Lem. 2. Thus, for every reachable monomial $\boldsymbol{x}^\beta$ we have

$$\deg_W(\boldsymbol{x}^\beta) \;\leq\; \deg_W(\boldsymbol{x}^\alpha).$$

Since all weights W_p are positive, this implies each block sum $s_p(\beta)$ is bounded:

$$0 \leq s_p(\beta) \leq \frac{\deg_W(\boldsymbol{x}^\alpha)}{W_p}, \qquad p = 1, \ldots, r.$$

In particular, each individual exponent is bounded:

$$0 \leq \beta_i \leq s_p(\beta) \leq \frac{\deg_W(\boldsymbol{x}^\alpha)}{W_p} \quad \text{for all } i \in G_p,\ p = 1, \ldots, r.$$

Therefore, the set of all multi-indices $\beta \in \mathbb{N}^n$ with $\deg_W(\boldsymbol{x}^\beta) \leq \deg_W(\boldsymbol{x}^\alpha)$ is finite, hence only finitely many monomials are reachable from $\boldsymbol{x}^\alpha$. Based on the above analysis, we present the main result for pro-solvable SDEs as follows.

Theorem 4. *If SDE (1) is pro-solvable, then Alg. 1 terminates after finitely many iterations for any monomial $\boldsymbol{x}^\alpha$, and SDE (1) is moment-solvable.*

Proof. By Lem. 2, only finitely many monomials can appear when iteratively executing lines 3–9 of Alg. 1, starting from any $\boldsymbol{x}^\alpha$. It implies that Alg. 1 terminates for any multi-index α, hence SDE (1) is moment-solvable by Thm. 3. □

Complexity Analysis. The computational complexity of our method for exact moment estimation arises primarily from two sources: constructing the moment closure set S in Alg. 1, and solving the resulting linear ODE system (3).

(1) Complexity of closure construction. For a fixed pro-solvable SDE and initial multi-index α, let S_α denote the finite set of monomials reachable from x^α by Alg. 1. By Lem. 2 and the argument preceding Thm. 4, there exists a constant C_0 (depending only on the SDE) such that for every $x^\beta \in S_\alpha$,

$$|\beta| \;=\; \sum_{i=1}^{n} \beta_i \;\leq\; \sum_{p=1}^{r} \frac{\deg_W(\boldsymbol{x}^\alpha)}{W_p} \;\leq\; C_0|\alpha|\,.$$

Hence, S_α is contained within the set of all monomials of total degree at most $C_0|\alpha|$ in n variables, which gives the combinatorial bound

$$|S_\alpha| \leq \binom{n + C_0|\alpha|}{C_0|\alpha|}.$$

In particular, for fixed dimension n, this yields $|S_\alpha| = O\big((|\alpha|)^n\big)$, while for fixed moment order $k = |\alpha|$, we have $|S_\alpha| = O\big(n^k\big)$. Since Alg. 1 processes each element of S_α at most once, the time and memory complexity of closure construction (cf. lines $3-9$ in Alg. 1) is $O(|S_\alpha|)$. The hidden constants in the big-O notation depend only on the given SDE, in particular on the degrees and number of monomials in the polynomial drift and diffusion coefficients.

(2) Complexity of solving the ODE system. According to Eq. (4) in Thm. 2, solving the resulting ODE system reduces to computing the matrix exponential e^{At} where A is a $|S_\alpha|$ dimensional matrix. Using standard algebraic methods (e.g., Jordan or rational canonical form), this can be done in time polynomial in $|S_\alpha|$, with worst-case complexity $\mathcal{O}(|S_\alpha|^3)$. ◁

5 Experiments

To demonstrate the effectiveness and applicability of our EME framework, we implemented [1] Alg. 1 in Python 3.13, leveraging standard symbolic and numerical linear algebra libraries. Given a polynomial SDE (1) and a target monomial x^α, our prototype automatically constructs the moment-closure set S_α, derives the corresponding linear ODE system, and computes the desired α-moment.

Benchmarks. We evaluated our method on a suite of SDE benchmarks (details see Appendix B), encompassing both linear and nonlinear examples in the literature as well as models with practical relevance. Specifically, we present in detail two cases: a consensus network with noise [20] and a nonlinear vehicle platoon system adapted from [14] to illustrate our method and demonstrate its usefulness for verification problems. All experiments were performed on a MacBook Pro with an Apple M4 processor, 16 GB of RAM, and running macOS Sequoia.

5.1 Case studies

Consensus network with noise [20]. This model describes an $n+1$ dimensional multi-agent consensus network (e.g., distributed sensors or robots) with noisy communication:

$$\mathrm{d}X_i(t) = \Big(a_i - \lambda X_i(t) + \kappa\big(X_{i+1}(t) - 2X_i(t) + X_{i-1}(t)\big)\Big)\,\mathrm{d}t + \sigma_i X_i(t)\,\mathrm{d}W_t^{(i)},$$

for $i = 1, \ldots, n+1$, with periodic boundary conditions $X_0(t) = X_n(t), X_{n+1}(t) = X_1(t)$. Since this dynamic is linear, it is pro-solvable by Def. 3. Consider the case $n = 2$ with parameter values $a_1 = 0$, $a_2 = 0$, $\lambda = 1$, $\kappa = 0.5$, $\sigma_1 = 1$, $\sigma_2 = 1$, and initial state $(1, 0)$. The verification objective is to ensure that, with probability at least $1 - e^{-t}$, the disagreement between the two agents, encoded by $|X_1(t) - X_2(t)|$, remains less than 0.1 for any $t \geq 10$, that is,

$$P\left(|X_1(t) - X_2(t)| \geq 0.1\right) \leq e^{-t} \quad \text{for } t \geq 10.$$

To this end, we compute $\mathbb{E}[(X_1(t) - X_2(t))^2]$. By explicitly calculating the moments $\mathbb{E}[X_1(t)^2]$, $\mathbb{E}[X_1(t)X_2(t)]$, and $\mathbb{E}[X_2^2(t)]$, we obtain

$$\mathbb{E}[(X_1(t) - X_2(t))^2] \;=\; \frac{(17 - 3\sqrt{17})e^{\frac{(\sqrt{17}-7)t}{2}} + (17 + 3\sqrt{17})e^{-\frac{(\sqrt{17}+7)t}{2}}}{34}.$$

Consequently, by applying Markov's inequality, we obtain, for any $t \geq 10$,

$$P\left(|X_1(t) - X_2(t)| \geq 0.1\right) \leq \frac{\mathbb{E}[(X_1(t) - X_2(t))^2]}{0.1^2} \leq e^{-t}\,.$$

This verifies the goal. It is worth noting that, to the best of our knowledge, this verification problem cannot be solved directly by a martingale-based approach that seeks a polynomial $h(t, x_1, x_2)$ satisfying the supermartingale condition to upper bound $\mathbb{E}[(X_1(t) - X_2(t))^2]$. Specifically, if such a polynomial

[1] Available at https://github.com/Shenghua-Feng/Exact_Moment_Estimation

h existed, we have $P(|X_1(t) - X_2(t)| \geq 0.1) \leq 100\,\mathbb{E}[(X_1(t) - X_2(t))^2] \leq 100\,\mathbb{E}[h(t, X_1(t), X_2(t))] \leq 100\,h(0, x_1(0), x_2(0))$. Since there must exist some $T > 10$ such that $e^{-T} < 100\,h(0, x_1(0), x_2(0))$, the standard martingale-based method cannot certify the property holds for all $t \geq 10$. ◁

Vehicle platoon [14]. We consider a nonlinear system adapted from [14] that involves two vehicles moving along a straight lane. For each vehicle, denote its position and velocity by $(p_1(t), v_1(t))$ and $(p_2(t), v_2(t))$, respectively. The first vehicle acts as the leader and follows a stochastic acceleration model:

$$\mathrm{d}p_1 = v_1\,\mathrm{d}t, \qquad \mathrm{d}v_1 = \big(-a_1 v_1 + u_1\big)\,\mathrm{d}t + \sigma_1\,\mathrm{d}W_t^{(1)},$$

where $a_1 > 0$ is a damping coefficient, u_1 is a control input (desired acceleration), and σ_1 scales the driving noise $W_t^{(1)}$. The second vehicle implements a nonlinear control law based on both its own velocity and the velocity of the leader:

$$\mathrm{d}p_2 = v_2\,\mathrm{d}t, \qquad \mathrm{d}v_2 = \big(-a_2 v_2 + (v_1 - 1)^2\big)\,\mathrm{d}t + \sigma_2\,\mathrm{d}W_t^{(2)},$$

where a_2 and σ_2 are parameters. It can be checked that this system is pro-solvable under the ordered partition $G_1 = \{v_1\}$, $G_2 = \{p_1, p_2, v_2\}$. Consider the parameter instantiation $a_1 = 1$, $u_1 = 1$, $\sigma_1 = 1$, $a_2 = 1$, and $\sigma_2 = 1$, with initial state $(p_1, v_1, p_2, v_2) = (1, 0, 0, 0)$. Suppose that the verification objective is to ensure that the expected distance between the vehicles, $\mathbb{E}[p_1(t) - p_2(t)]$, always remains between 0.5 and 1.5 for all $t \geq 0$.

To verify this, we compute the expected values $\mathbb{E}[p_1(t)]$ and $\mathbb{E}[p_2(t)]$, yielding

$$\frac{3}{4} \leq \mathbb{E}\big[p_1(t) - p_2(t)\big] = \frac{3}{4} + \frac{e^{-t}}{2} - \frac{e^{-2t}}{4} \leq 1,$$

which verifies the desired safety property. ◁

5.2 Evaluation of effectiveness

Table 1 summarizes the experimental results of our method on a diverse suite of polynomial SDE benchmarks, which cover a range of system dimensions, polynomial degrees, and moment orders, illustrating the generality of our approach.

Efficiency and scalability. For all pro-solvable SDEs, our method successfully constructs the finite closures and computes the exact moment for all tested cases. Closure construction times are consistently short, and the dimension $|S_\alpha|$ scales polynomially with the moment order and system size (cf. ou-env), consistent with our theoretical analysis. The subsequent ODE solving is also efficient for moderate dimensions, with larger $|S_\alpha|$ (e.g., high-order moments in gene) leading to higher computational cost primarily due to the complexity of matrix exponentiation. Note that developing more efficient symbolic solvers for linear ODEs is a complementary and orthogonal direction to our work; in our implementation, we simply rely on off-the-shelf symbolic packages for this step.

Table 1: Experimental results for exact moment calculation.

	SDE System			Moment		Obtained Closure S_α			Solve ODE	
Benchmark	dim	deg	p-s	$\boldsymbol{x}^\alpha$	$\lvert\alpha\rvert$	succ	time	$\lvert S_\alpha\rvert$	succ	time
ou-env [13]	2	2	yes	$\mathbb{E}[x_2^2]$	2	✓	0.01s	8	✓	0.2s
				$\mathbb{E}[x_2^3]$	3	✓	0.02s	15	✓	0.6s
				$\mathbb{E}[x_2^4]$	4	✓	0.02s	24	✓	1.2s
				$\mathbb{E}[x_2^5]$	5	✓	0.04s	35	✓	2.8s
				$\mathbb{E}[x_2^{10}]$	10	✓	0.17s	120	✓	39.4s
gene [24]	5	3	yes	$\mathbb{E}[x_1x_5]$	2	✓	0.04s	23	✓	3.0s
				$\mathbb{E}[x_5^2]$	2	✓	0.14s	85	✓	79.6s
				$\mathbb{E}[x_1x_5^2]$	3	✓	0.17s	115	✓	164.1s
consensus [20]	2	1	yes	$\mathbb{E}[x_1x_2]$	2	✓	0.01s	3	✓	0.2s
vehicles [14]	4	2	yes	$\mathbb{E}[x_2^2]$	2	✓	0.01s	13	✓	0.5s
oscillator [11]	3	2	yes	$\mathbb{E}[x_2x_3^2]$	3	✓	0.01s	6	✓	2.3s
coupled3d	3	3	no	$\mathbb{E}[x_1^2x_2^2]$	4	✓	0.01s	3	✓	0.2s

dim: Dimension of the SDE system; **deg**: Maximum polynomial degree of drift/diffusion terms in the SDE; **p-s**: Whether the SDE is pro-solvable; $\boldsymbol{x}^\alpha$: Target moment to compute; $|\alpha|$: Degree of the target moment; **succ**: Whether a closed linear ODE system was successfully constructed and solved; **time**: Time required to obtain the closure S_α (i.e., construct or solve the linear ODE system); $|\mathbf{S}_\alpha|$: Dimension of the resulting linear ODE system.

Comparison across models. Linear and low-dimensional systems (such as the consensus and oscillator) exhibit particularly fast closure and solution times. For nonlinear pro-solvable examples (e.g., `ou-env` and `gene`), the closure remains tractable even for moments of degree up to 10, validating the practical scalability of our framework. The benchmark coupled3d further shows that our method may still terminate for certain SDEs that *do not* satisfy the pro-solvable property; however, termination is not guaranteed in general.

Overall, the experimental results demonstrate that our approach is effective, broadly applicable to both linear and a wide class of nonlinear systems, and scales well in practice for pro-solvable SDEs.

6 Conclusion

We presented a generic symbolic method for exact moment estimation of polynomial SDEs, and identified a broad class of pro-solvable systems in which all moments can be computed exactly via finite-dimensional linear ODEs. Both theoretical analysis and experimental results demonstrate its effectiveness and scalability for a wide range of linear and nonlinear models, paving the way for moment-based verification and analysis of stochastic dynamical systems. As future work, we plan to characterize more subclasses of moment-solvable SDEs.

References

1. Armstrong, T.B., Kolesár, M.: Sensitivity analysis using approximate moment condition models. Quantitative Economics **12**(1), 77–108 (2021)
2. Bartocci, E., Kovács, L., Stankovič, M.: Automatic generation of moment-based invariants for prob-solvable loops. In: International Symposium on Automated Technology for Verification and Analysis. pp. 255–276. Springer (2019)
3. Black, F., Scholes, M.: The pricing of options and corporate liabilities. Journal of Political Economy **81**(3), 637–654 (1973)
4. Borri, A., Carravetta, F., Palumbo, P.: Cubification of $\sigma\pi$-sde and exact moment equations. Systems & Control Letters **136**, 104602 (2020)
5. Feng, S., Chen, M., Xue, B., Sankaranarayanan, S., Zhan, N.: Unbounded-time safety verification of stochastic differential dynamics. In: International Conference on Computer Aided Verification. pp. 327–348. Springer (2020)
6. Gardiner, C.W., et al.: Handbook of stochastic methods, vol. 3. springer Berlin (2004)
7. Ghusinga, K.R., Lamperski, A., Singh, A.: Moment analysis of stochastic hybrid systems using semidefinite programming. Automatica **112**, 108634 (2020)
8. Ghusinga, K.R., Soltani, M., Lamperski, A., Dhople, S.V., Singh, A.: Approximate moment dynamics for polynomial and trigonometric stochastic systems. In: 2017 IEEE 56th Annual Conference on Decision and Control (CDC). pp. 1864–1869. IEEE (2017)
9. Guidoum, A.C., Boukhetala, K.: Exact higher-order moments for linear nonhomogeneous stochastic differential equation. Model Assisted Statistics and Applications **18**(4), 321–329 (2023)
10. Gunawan, R., Cao, Y., Petzold, L., Doyle, F.J.: Sensitivity analysis of discrete stochastic systems. Biophysical journal **88**(4), 2530–2540 (2005)
11. Hafstein, S., Gudmundsson, S., Giesl, P., Scalas, E., et al.: Lyapunov function computation for autonomous linear stochastic differential equations using sum-of-squares programming. DISCRETE AND CONTINUOUS DYNAMICAL SYSTEMS. SERIES B. **23**(2), 939–956 (2018)
12. Hoogendoorn, S., Bovy, P.: Pedestrian route-choice and activity scheduling theory and models. Transportation Research Part B: Methodological **38**(2), 169–190 (2004)
13. Kallianpur, G., Xiong, J.: Stochastic models of environmental pollution. Advances in Applied Probability **26**(2), 377–403 (1994)
14. Kavathekar, P., Chen, Y.: Vehicle platooning: A brief survey and categorization. In: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. vol. 54808, pp. 829–845 (2011)
15. Lamperski, A., Ghusinga, K.R., Singh, A.: Analysis and control of stochastic systems using semidefinite programming over moments. IEEE Transactions on Automatic Control **64**(4), 1726–1731 (2018)
16. Lasserre, J.B.: The moment-sos hierarchy. In: Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018. pp. 3773–3794. World Scientific (2018)
17. Lee, C.H., Kim, K.H., Kim, P.: A moment closure method for stochastic reaction networks. The Journal of chemical physics **130**(13) (2009)
18. Moosbrugger, M., Stankovič, M., Bartocci, E., Kovács, L.: This is the moment for probabilistic loops. Proceedings of the ACM on Programming Languages **6**(OOPSLA2), 1497–1525 (2022)

19. Øksendal, B.: Stochastic Differential Equations: An introduction with applications. Springer Science & Business Media (2013)
20. Olfati-Saber, R., Fax, J.A., Murray, R.M.: Consensus and cooperation in networked multi-agent systems. Proceedings of the IEEE **95**(1), 215–233 (2007)
21. Panik, M.: Stochastic Differential Equations: An Introduction with Applications in Population Dynamics Modeling. Wiley (2017)
22. Prajna, S., Jadbabaie, A., Pappas, G.J.: A framework for worst-case and stochastic safety verification using barrier certificates. IEEE Transactions on Automatic Control **52**(8), 1415–1428 (2007)
23. Scarciotti, G., Teel, A.R.: On moment matching for stochastic systems. IEEE Transactions on Automatic Control **67**(2), 541–556 (2021)
24. Sinigh, A., Hespanha, J.P.: Stochastic analysis of gene regulatory networks using moment closure. In: 2007 American Control Conference. pp. 1299–1304. IEEE (2007)
25. Sontag, E.D.: Examples of computation of exact moment dynamics for chemical reaction networks. In: Emerging Applications of Control and Systems Theory: A Festschrift in Honor of Mathukumalli Vidyasagar, pp. 295–312. Springer (2018)
26. Sontag, E.D., Singh, A.: Exact moment dynamics for feedforward nonlinear chemical reaction networks. IEEE Life Sciences Letters **1**(2), 26–29 (2015)
27. Williams, D.: Probability with martingales. Cambridge university press (1991)

A Details for Example 1

In Exmp. 1, we obtain a closed 8-dimensional linear ODE system for the collection of moments

$$m_{(0,2)},\quad m_{(2,1)},\quad m_{(2,0)},\quad m_{(1,1)},\quad m_{(4,0)},\quad m_{(3,0)},\quad m_{(0,1)},\quad m_{(1,0)}$$

where $m_{(i,j)} \triangleq \mathbb{E}[X_t^i Y_t^j]$. The corresponding ODE system is

$$\begin{aligned}
\dot{m}_{(0,2)}(t) &= -4m_{(0,2)}(t) + 2m_{(2,1)}(t) + m_{(2,0)}(t) + 2m_{(1,1)}(t)\\
\dot{m}_{(2,1)}(t) &= -4m_{(2,1)}(t) + m_{(4,0)}(t) + m_{(3,0)}(t) + m_{(0,1)}(t)\\
\dot{m}_{(2,0)}(t) &= 1 - 2m_{(2,0)}(t)\\
\dot{m}_{(1,1)}(t) &= m_{(2,0)}(t) - 3m_{(1,1)}(t) + m_{(3,0)}(t)\\
\dot{m}_{(4,0)}(t) &= 6m_{(2,0)}(t) - 4m_{(4,0)}(t)\\
\dot{m}_{(3,0)}(t) &= -3m_{(3,0)}(t) + 3m_{(1,0)}(t)\\
\dot{m}_{(0,1)}(t) &= m_{(2,0)}(t) - 2m_{(0,1)}(t) + m_{(1,0)}(t)\\
\dot{m}_{(1,0)}(t) &= -m_{(1,0)}(t)
\end{aligned}$$

Solving this linear ODE system yields the explicit expression

$$\mathbb{E}\left[Y_t^2\right] = m_{(0,2)}(t) = \frac{1}{3} + \frac{2}{3}e^{-3t} + \left(-\frac{t}{4} - \frac{11}{8}\right)e^{-2t} + \left(\frac{3}{4}t^2 + t + \frac{3}{8}\right)e^{-4t}.$$

B Benchmarks

Benchmark 1 (ou-env [13]) *The system dynamics is the same as in Exmp. 1:*

$$\begin{cases} \mathrm{d}X_t = -X_t\,\mathrm{d}t + \mathrm{d}W_t^{(1)},\\ \mathrm{d}Y_t = \left(-2Y_t + X_t + X_t^2\right)\mathrm{d}t + X_t\,\mathrm{d}W_t^{(2)}, \end{cases} \tag{11}$$

with initial state $(X_0, Y_0) = (0, 0)$. ◁

Benchmark 2 (gene [24]) *The system dynamics is:*

$$\begin{cases}
\mathrm{d}X_{1,t} = \left(-X_{1,t} + 1\right)\mathrm{d}t + 0.5\,\mathrm{d}W_t^{(1)},\\
\mathrm{d}X_{2,t} = \left(1.2\,X_{1,t} - 0.8\,X_{2,t}\right)\mathrm{d}t + \left(0.3\,X_{1,t} + 0.4\right)\mathrm{d}W_t^{(2)},\\
\mathrm{d}X_{3,t} = \left(1.0\,X_{2,t} - 0.7\,X_{3,t} + 0.2\,X_{1,t}^2\right)\mathrm{d}t + \left(0.5\,X_{2,t} + 0.1\,X_{1,t}^2\right)\mathrm{d}W_t^{(3)},\\
\mathrm{d}X_{4,t} = \left(0.9\,X_{3,t} - 0.6\,X_{4,t} + 0.1\,X_{1,t}X_{2,t}\right)\mathrm{d}t + \left(0.4\,X_{3,t} + 0.2\,X_{2,t}^2\right)\mathrm{d}W_t^{(4)},\\
\mathrm{d}X_{5,t} = \left(0.8\,X_{4,t} - 0.5\,X_{5,t} + 0.15\,X_{3,t}^2 + 0.05\,X_{1,t}^3\right)\mathrm{d}t\\
\qquad + \left(0.3\,X_{4,t} + 0.1\,X_{3,t}^2 + 0.05\,X_{1,t}^3\right)\mathrm{d}W_t^{(5)}.
\end{cases} \tag{12}$$

with initial state $X_{i,0} = 0$ *for* $i = 1, 2, \ldots, 5$. ◁

Benchmark 3 (consensus [20]) *The system dynamics correspond to those in the first case study, namely the consensus network with noise. Under the specific parameter instantiation considered there, the dynamics are given by*

$$\begin{cases} \mathrm{d}X_{1,t} = \big(-2X_{1,t} + X_{2,t}\big)\,\mathrm{d}t + \, X_{1,t}\,\mathrm{d}W_t^{(1)}, \\ \mathrm{d}X_{2,t} = \big(X_{1,t} - 2X_{2,t}\big)\,\mathrm{d}t + \, X_{2,t}\,\mathrm{d}W_t^{(2)}. \end{cases} \tag{13}$$

with initial state $(X_{1,0}, X_{2,0}) = (1, 0)$. ◁

Benchmark 4 (vehicles [14]) *The system dynamics correspond to those in the second case study, namely the vehicle platoon. Under the specific parameter instantiation considered there, the dynamics are given by*

$$\begin{cases} \mathrm{d}p_1 = v_1\,\mathrm{d}t, \\ \mathrm{d}v_1 = \big(-v_1 + 1\big)\,\mathrm{d}t + \,\mathrm{d}W_t^{(1)}, \\ \mathrm{d}p_2 = v_2\,\mathrm{d}t, \\ \mathrm{d}v_2 = \big(-v_2 + (v_1 - 1)^2\big)\,\mathrm{d}t + \,\mathrm{d}W_t^{(2)}. \end{cases} \tag{14}$$

with initial state $(p_1, v_1, p_2, v_2) = (1, 0, 0, 0)$. ◁

Benchmark 5 (oscillator [11]) *The system dynamics is:*

$$\begin{cases} \mathrm{d}X_{1,t} = X_{2,t}\,\mathrm{d}t, \\ \mathrm{d}X_{2,t} = \big(-0.3\,X_{2,t} - X_{1,t} + 0.8\,X_{3,t}^2\big)\,\mathrm{d}t + 0.2\,X_{2,t}\,\mathrm{d}W_t^{(1)}, \\ \mathrm{d}X_{3,t} = -X_{3,t}\,\mathrm{d}t + 0.5\,\mathrm{d}W_t^{(2)}, \end{cases} \tag{15}$$

with initial state $X_{i,0} = 0$ *for* $i = 1, 2, 3$. ◁

Benchmark 6 (coupled3d) *The system dynamics is:*

$$\begin{cases} \mathrm{d}X_{1,t} = \big(-\frac{1}{2}X_{1,t} - X_{1,t}X_{2,t} - \frac{1}{2}X_{1,t}X_{2,t}^2\big)\,\mathrm{d}t + X_{1,t}\big(1 + X_{2,t}\big)\,\mathrm{d}W_t^{(1)} \\ \mathrm{d}X_{2,t} = \big(-X_{2,t} + X_{3,t}\big)\,\mathrm{d}t + 0.3\,X_{3,t}\,\mathrm{d}W_t^{(2)}, \\ \mathrm{d}X_{3,t} = \big(X_{2,t} - X_{3,t}\big)\,\mathrm{d}t + 0.3\,X_{2,t}\,\mathrm{d}W_t^{(3)}. \end{cases} \tag{16}$$

with initial state $(X_{1,0}, X_{2,0}, X_{3,0}) = (0, 0, 0)$. ◁